

# ESTRUCTURA DE DATOS DINÁMICOS. UNA FORMA FÁCIL DE APRENDER

7-9 minutos

---

## ÍNDICE

SOBRE LOS AUTORES

SOBRE LOS AUTORES

PREFACIO

PRÓLOGO

CAPÍTULO 1. INTRODUCCIÓN A LAS ESTRUCTURAS DE  
DATOS

1.1 DEFINICIÓN DE ESTRUCTURA DE DATOS

1.2 OPERACIONES

1.3 CLASIFICACIÓN

1.3.1 Estructuras de Datos Internas y Externas.

1.3.2 Estructuras de Datos Estáticas.

1.3.3 Estructuras de Datos Dinámicas.

1.3.4 Estructuras de Datos Lineales

1.3.5 Estructuras de Datos no Lineales.

CAPÍTULO 2. ANÁLISIS DE ALGORITMOS.

2.1 ALGUNAS ACLARACIONES.

2.2 DEFINICIÓN DE ALGORITMO

2.3 FACTORES QUE INFLUYEN EN LA EFICIENCIA DE UN  
ALGORITMO.

2.4 COMPLEJIDAD COMPUTACIONAL

2.5 FUNCIÓN DE COMPLEJIDAD.

2.6 PRELIMINARES MATEMÁTICOS.

2.6.1 Propiedades de sumatorias

2.6.2 Funciones piso y techo.

2.7 CÁLCULO DE LA FUNCIÓN DE COMPLEJIDAD.

2.7.1 Calcular  $T(n)$  para sentencias consecutivas

2.7.2 Calcular  $T(n)$  para condicionales

8 ESTRUCTURAS DE DATOS DINÁMICAS © RA-MA

2.7.3 Calcular  $T(n)$  para ciclos.

2.8 ORDEN DE MAGNITUD (NOTACIÓN O GRANDE)

2.8.1 Definición matemática del Orden de Magnitud

CAPÍTULO 3. TIPOS ABSTRACTOS DE DATOS

3.1 INTRODUCCIÓN.

3.2 DEFINICIÓN DE TAD

3.3 MÉTODOS PARA ESPECIFICAR UN TAD

3.3.1 Métodos formales.

3.3.2 Métodos no formales

3.3.3 Métodos semiformales.

3.4 CLASIFICACIÓN DE LAS OPERACIONES DE UN TAD.

3.5 EJERCICIOS PROPUESTOS

CAPÍTULO 4. PILAS

4.1 INTRODUCCIÓN.

4.2 DEFINICIÓN DE PILA.

4.3 EL TAD PILA

4.4 IMPLEMENTACIÓN DEL TAD PILA EN JAVA

4.5 LA CLASE STACK DE JAVA.

4.6 PROBLEMAS QUE SE RESUELVEN CON PILAS.

4.6.1 Evaluación de la correspondencia de delimitadores

4.6.2 Evaluación de expresiones aritméticas

## 4.7 EJERCICIOS PROPUESTOS

## CAPÍTULO 5. COLAS

### 5.1 INTRODUCCIÓN.

### 5.2 DEFINICIÓN DE COLA.

### 5.3 EL TAD COLA

### 5.4 IMPLEMENTACIÓN DEL TAD COLA EN JAVA

### 5.5 LA INTEFACE QUEUE DE JAVA

### 5.6 PROBLEMAS QUE SE RESUELVEN CON COLAS.

#### 5.6.1 Simulador del despegue y aterrizaje de aviones.

## CAPÍTULO 6. LISTAS.

### 6.1 INTRODUCCIÓN.

### 6.2 DEFINICIÓN DE LISTA

### 6.3 TIPOS DE LISTAS

#### 6.3.1 Listas enlazadas simples.

#### 6.3.2 Listas doblemente enlazadas

#### 6.3.3 Listas Circulares

#### 6.3.4 Listas doblemente enlazadas circulares.

#### 6.3.5 Listas de listas.

### © RA-MA ÍNDICE 9

### 6.4 EL TAD LISTA.

### 6.5 IMPLEMENTACIÓN DEL TAD LISTA EN JAVA.

### 6.6 LAS CLASES ARRAYLIST Y LINKEDLIST DE JAVA

### 6.7 PROBLEMAS QUE SE RESUELVEN CON LISTAS

#### 6.7.1 Lista de contactos personales

#### 6.7.2 Préstamo de recursos.

### 6.8 LISTA DOBLEMENTE ENLAZADA

#### 6.8.1 Implementación de la Lista Doblemente Enlazada.

### 6.9 LISTA CIRCULAR.

#### 6.9.1 Implementación de la Lista Circular

### 6.10 LISTA DOBLEMENTE ENLAZADA CIRCULAR

6.10.1 Implementación de la Lista Doblemente Enlazada Circula

6.11 EJERCICIOS PROPUESTOS.

## CAPÍTULO 7. ARBOLES

7.1 INTRODUCCIÓN.

7.2 CONCEPTOS Y TERMINOLOGÍA.

7.3 ARBOLES BINARIOS.

7.3.1 Definición de árbol binario

7.3.2 Equilibrio.

7.3.3 Árboles binarios completos.

7.3.4 TAD árbol binario

7.3.5 Operaciones en árboles binarios

7.3.6 Estructura de un árbol binario.

7.3.7 Representación de un nodo

7.3.8 Creación de un árbol binario.

7.3.9 Árbol de expresión.

7.3.10 Recorrido de un árbol.

7.3.11 Recorrido en preorden.

7.3.12 Recorrido en orden

7.3.13 Recorrido postorden.

7.3.14 Implementación de los recorridos

7.4 RECORRIDOS ITERATIVOS EN ÁRBOLES BINARIOS

7.4.1 Recorrido preorden iterativo.

7.4.2 Recorrido inorden iterativo

7.4.3 Recorrido postorden iterativo

7.5 RECORRIDO POR NIVELES EN ÁRBOLES BINARIOS

7.6 RESUMEN DE LOS RECORRIDOS DE ÁRBOLES BINARIOS

7.7 ÁRBOL BINARIO DE BÚSQUEDA

Ejercicio 7.1

7.7.1 Nodo de un árbol binario de búsqueda.

7.7.2 Operaciones en árboles binarios de búsqueda

7.7.3 Insertar un nodo.

## 10 ESTRUCTURAS DE DATOS DINÁMICAS © RA-MA

7.7.4 Implementación iterativa de la inserción.

7.7.5 Implementación recursiva de la inserción.

7.7.6 Búsqueda.

7.7.7 Implementación iterativa de la búsqueda

7.7.8 Implementación recursiva de la búsqueda

7.7.9 Eliminar un nodo.

7.7.10 Implementación iterativa de la eliminación

7.7.11 Resumen del Árbol binario de búsqueda.

## 7.8 ÁRBOLES DE BÚSQUEDA EQUILIBRADOS.

7.8.1 Eficiencia en la búsqueda de un árbol equilibrado

7.8.2 Árboles AVL

7.8.3 Inserción en árboles AVL.

7.8.4 Proceso de inserción de un nuevo nodo

7.8.5 Borrado de un nodo en un árbol equilibrado.

## 7.9 ÁRBOLES N-ARIOS

7.9.1 Definiciones y conceptos básicos.

7.9.2 El TAD ARBOLN

7.9.3 Alternativas de implementación del TAD ArbolN

7.9.4 Implementación del TAD ArbolN

## 7.10 EL ÁRBOL 1-2-3: UN ÁRBOL TRIARIO ORDENADO.

7.10.1 TAD árbol 1-2-3.

7.10.2 Implementación del árbol 1-2-3

7.10.3 Proceso de inserción

7.10.4 Proceso de eliminación

## 7.11 ÁRBOL 2-3: UN ÁRBOL TRIARIO ORDENADO

7.11.1 Definiciones y conceptos básicos

7.11.2 Un árbol B

7.11.3 Especificación del TAD árbol 2-3

7.11.4 Algoritmo de inserción

7.11.5 Algoritmo de eliminación

7.12 EL TAD TRIE: CONJUNTO DE PALABRAS.

7.12.1 TAD Trie

7.13 CUADTREE: REPRESENTACIÓN DE IMÁGENES

7.13.1 TAD cuadtrees

7.13.2 Implementación del TAD cuadtrees.

7.14 EJERCICIOS PROPUESTOS.

CAPÍTULO 8. GRAFOS.

8.1 INTRODUCCIÓN.

8.2 CONCEPTOS Y DEFINICIONES.

8.2.1 Grado de entrada, grado de salida de un grafo

8.2.2 Camino.

8.3 EL TAD GRAFO.

© RA-MA ÍNDICE 11

8.4 REPRESENTACIÓN DE LOS GRAFOS.

8.4.1 Matriz de adyacencia.

8.4.2 Implementación de la Matriz de Adyacencia

8.4.3 Lista de adyacencia

8.4.4 Implementación de la Lista de Adyacencia.

8.5 RECORRIDOS DE UN GRAFO.

8.5.1 Recorrido en anchura.

8.5.2 Recorrido en profundidad.

8.5.3 Implementación del recorrido en anchura y profundidad

8.6 CONEXIONES EN UN GRAFO.

8.6.1 Componentes conexas de un grafo

8.6.2 Matriz de caminos, cierre transitivo.

8.6.3 Matriz de caminos y cierre transitivo

8.7 MATRIZ DE CAMINOS: ALGORITMO DE WARSHALL.

8.7.1 Implementación del algoritmo de Warshall.

## 8.8 ALGORITMO DE COSTOS MÍNIMOS: DIJKSTRA

### 8.8.1 Implementación del Algoritmo de Dijkstra.

## 8.9 ALGORITMO DE FLOYD.

### 8.9.1 Implementación del Algoritmo de Floyd

## 8.10 EJERCICIOS PROPUESTOS.

## REFERENCIAS

## ÍNDICE ALFABÉTICO

## **SINOPSIS**

Las estructuras de datos son importantes en el desarrollo de aplicaciones informáticas. Cuando se construyen los primeros programas, es decir, programas pequeños y sencillos, se piensa directamente en la escritura del código. Sin embargo, conforme los programas se hacen más largos y complejos, es necesario, poner atención a otros aspectos del software además de la codificación. Si se es un profesional del software, se requiere trabajar como parte de un equipo que desarrolla un sistema que contiene miles o incluso millones de líneas de código.

En estos escenarios temas como la programación orientada a objetos, patrones de diseño, el proceso de desarrollo, buenas prácticas de programación, buenas prácticas de comunicación, ciclos de vida del software y, por supuesto, estructuras de datos adecuadas que manipulen los datos, cobran vital importancia. Las aplicaciones actuales son complejas y se escriben en capas, donde cada capa maneja un tipo de lógica específica. Los datos son una capa importante en cualquier aplicación, y su manipulación adecuada depende de las estructuras de datos que utilicemos.

Este libro enseña de manera didáctica las estructuras de datos

dinámicas lineales y no lineales: pilas, colas, listas, árboles y grafos. Se utiliza un pensamiento abstracto para estudiar cada estructura de datos, por ello se propone el TAD (Tipo Abstracto de Datos), su implementación y la resolución de problemas reales donde se pone en uso las estructuras de datos.

El libro utiliza un paradigma orientado a objetos y ejemplos de implementaciones en lenguaje Java. Los algoritmos que manipulan dichas estructuras de datos también son tenidos en cuenta desde el punto de vista de la complejidad computacional. Por ello, en sus primeros capítulos se introduce en las técnicas básicas de análisis de algoritmos para calcular la eficiencia de los algoritmos.

El libro contribuye a la formación de estudiantes de ingeniería de sistemas, ingeniería informática y profesionales afines, que se inician en el fascinante mundo de la construcción de aplicaciones orientadas a objetos de mayor complejidad.